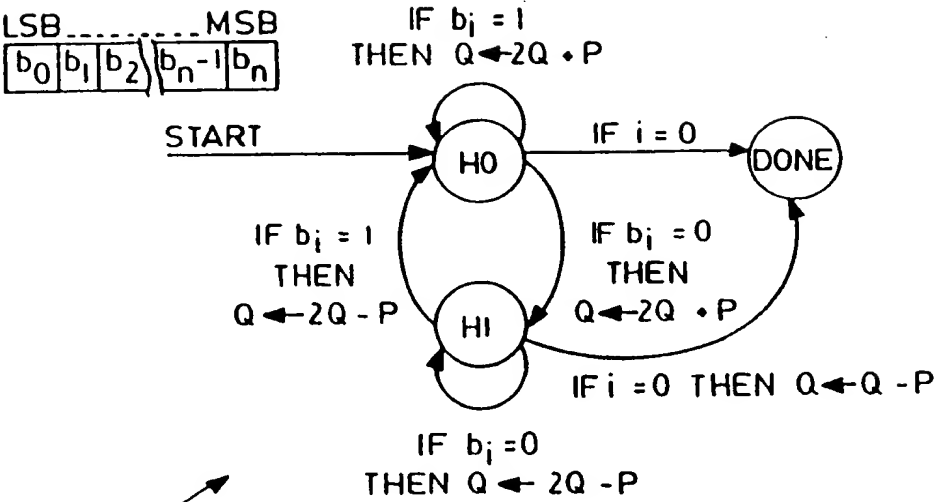


FIG. 1

$$\begin{aligned}
 629 &= \begin{array}{c} \begin{array}{cccccccc} 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^2 & 2^0 \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ \hline \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline \cdot 1 & \cdot 1 & -1 & -1 & \cdot 1 & \cdot 1 & \cdot 1 & -1 \\ \hline \end{array} \\ \hline &= 2^9 \cdot (2^8 - 2^7 - 2^6) \cdot 2^5 \cdot 2^4 \cdot (2^3 - 2^2) \cdot (2^1 - 2^0)
 \end{array} \\
 628 &= \begin{array}{c} \begin{array}{cccccccc} 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^2 & 2^0 \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ \hline \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline \cdot 1 & \cdot 1 & -1 & -1 & \cdot 1 & \cdot 1 & \cdot 1 & -1 \\ \hline \end{array} \\ \hline &= 2^9 \cdot (2^8 - 2^7 - 2^6) \cdot 2^5 \cdot 2^4 \cdot (2^3 - 2^2) \cdot (2^1 - 2^0)
 \end{array}
 \end{aligned}$$

FIG. 2



STATE	INPUT	NEXT	ACTION
H0	i < 0	DONE	SUBTRACT
H0	b <sub>i</sub> = 0	H1	DOUBLE, ADD
H0	b <sub>i</sub> = 1	H0	DOUBLE, ADD
H1	i < 1	DONE	
H1	b <sub>i</sub> = 0	H1	DOUBLE, SUBTRACT
H1	b <sub>i</sub> = 1	H0	DOUBLE, SUBTRACT

FIG. 3

```

BEGIN :
    i := N          ; START FROM MSB          L 1
    Q := 0          ; INITIALIZE ACCUMULATOR   L 2
    H := 0          ; INITIALIZE STATE         L 3

LOOP :          ; FOR ALL BITS
    Q := Q + Q      ; DOUBLE ACCUMULATOR      L 4

    IF H = 0        ; IF H STATE IS SET        L 5
        Q := Q + P  ; ADD BASE POINT TO ACCUMULATOR L 6
        GOTO ENDLOOP ;                      L 7
    ELSE           ; ELSE
        Q := Q + (-P) ; SUBTRACT BASE POINT      L 8
        GOTO ENDLOOP ;                      L 9

ENDLOOP :
    H := b[i]       ; SET H STATE TO VALUE OF b[i]   L 10
    i := i - 1      ; PROCESS NEXT BIT              L 11
    IF i > 0         ; IF BIT EXISTS                L 12
        GOTO LOOP   ; CONTINUE AT TOP OF LOOP        L 13

    IF H = 0        ; IF EXISTING FROM H = 0 STATE    L 14
        Q := Q + (-P) ; CORRECT RESULT BY FINAL SUBTRACT L 15
    END             ;                      L 16

```

FIG. 4

5/7

BEGIN :  
     i := N ; START FROM MSB LL1  
     Q := 0 ; INITIALIZE ACCUMULATOR LL2

H0 :  
     Q := Q + Q ; DOUBLE ACCUMULATOR LL3  
     Q := Q + P ; ADD BASE POINT TO ACCUMULATOR LL4  
     GOTO ENDLOOP ; BRANCH TO END OF LOOP TESTS LL5

H1 :  
     Q := Q + Q ; DOUBLE ACCUMULATOR LL6  
     Q := Q + (-P) ; SUBTRACT BASE POINT FROM LL7  
                     ACCUMULATOR  
     GOTO ENDLOOP ; BRANCH TO END OF LOOP TESTS LL8

ENDLOOP :  
     IF b[i]=1 ; IF CURRENT BIT IS SET LL9  
         GOTO NEXT H0 ; FOLLOW H0 PATH LL10  
                     ; ELSE FALL INTO H1 PATH

NEXT H1 :  
     ; H1 PATH  
     i := i - 1 ; PROCESS NEXT BIT LL11  
     IF i > 0 ; IF BIT EXISTS LL12  
         GOTO H1 ; EXECUTE H1 STATE LL13  
     Q := Q + (-P) ; ELSE CORRECT RESULT AND END LL14  
     END LL15

NEXT H0 :  
     ; H0 PATH  
     i := i - 1 ; PROCESS NEXT BIT LL16  
     IF i > 0 ; IF BIT EXISTS LL17  
         GOTO H0 ; EXECUTE H0 STATE LL18  
     END ; ELSE END LL19

FIG. 5

6/7

```
BEGIN :  
    i := N  
    Q := 1  
  
H0 :  
    Q := Q · Q ( $Q^2$ )  
    Q := Q · M  
    GOTO ENDLOOP  
  
H1 :  
    Q := Q · Q  
    Q := Q / M ( $Q \cdot M^{-1}$ )  
  
ENDLOOP :  
    IF b[i] = 1 GOTO ENDLOOP  
  
NEXT H1 :  
    i = i - 1  
    IF i > 0  
        GOTO H1  
    Q = Q / M  
    END  
  
NEXT H0 :  
    i = i - 1  
    IF i > 0  
        GOTO H0  
    END
```

60

FIG. 6

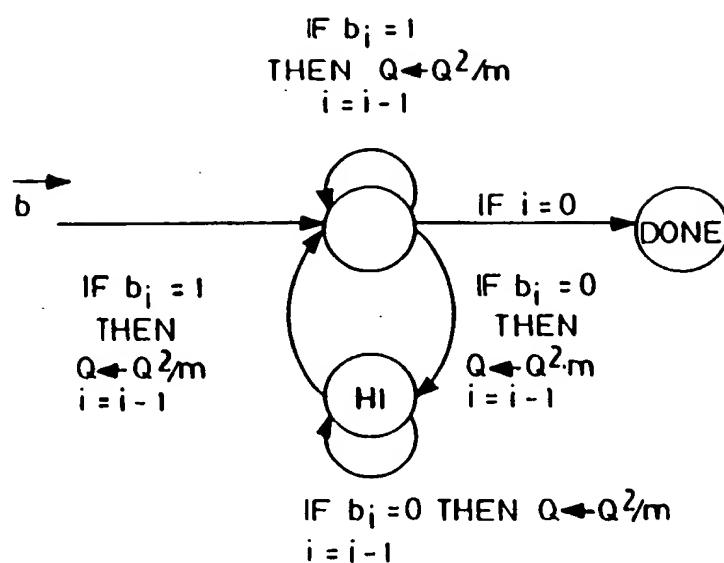


FIG. 7

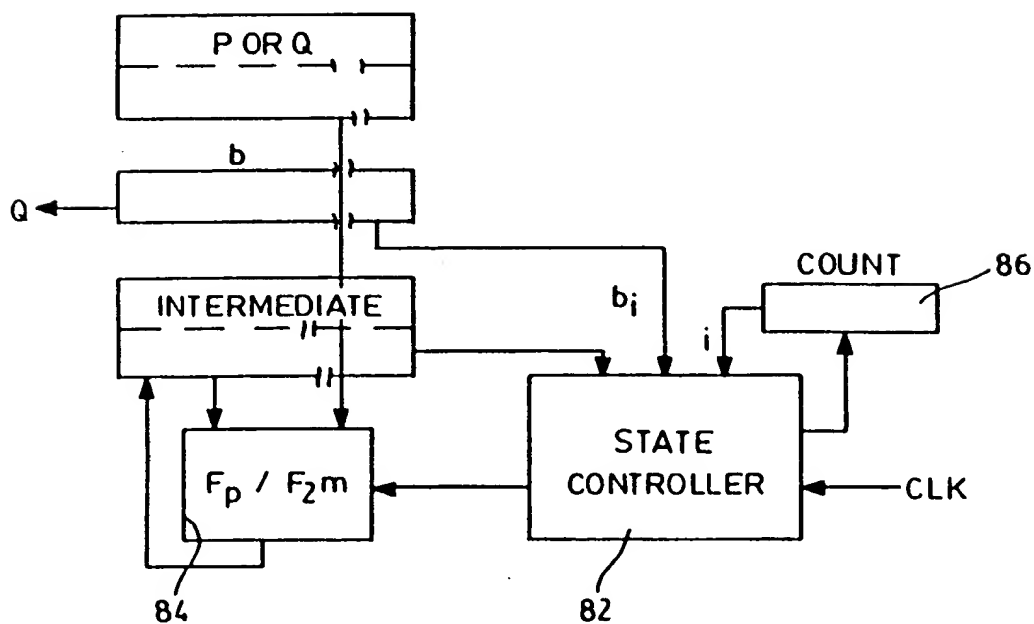


FIG. 8

2000